

CLAIMS

What is Claimed is:

- 5
1. A method for executing a dynamically loaded program, said program including a main program unit, said method comprising:  
 executing said main program unit a first time;  
 creating at least one library file containing only application program files loaded  
 during said first execution of said main program unit;  
 specifying a system program file input; and  
 10 executing said main program unit a second time using said system program file input  
 and said at least one library file for dynamically loaded program files.
  2. The method of claim 1 wherein said creating further comprises:  
 specifying a first at least one pathname for system program files;  
 15 specifying a second at least one pathname for application program files;  
 executing said main program unit using said first at least one pathname and said  
 second at least one pathname for dynamically loaded program files; and  
 storing each application program file loaded during execution of said main program  
 unit to said library file.
  - 20 3. The method of claim 2 wherein said storing further comprises:  
 loading a program file when referenced during execution of said main program unit;

storing said program file to said library file when said program file is an application  
program file; and  
determining whether execution of said main program unit has terminated.

5 4. The method of claim 3 wherein said library file further comprises a compressed file.

5. The method of claim 1 wherein said program files comprise Java™ class files and  
Java™ archive files.

6. The method of claim 5 wherein said system program file input comprises the Java™  
Development Kit.

7. The method of claim 6 wherein said library file comprises a Java™ archive file.

8. A method for optimizing a dynamically loaded program, said program including a  
main program unit, said method comprising:

creating at least one library file containing only application program files loaded  
during execution of said main program unit; and

optimizing said program based upon a list of application program files in said library  
file.

9. The method of claim 8 wherein said creating further comprises:  
specifying a first at least one pathname for system program files;  
specifying a second at least one pathname for application program files;  
executing said main program unit using said first at least one pathname and said  
5 second at least one pathname for dynamically loaded program files; and  
storing each application program file loaded during execution of said main program  
unit to said library file.

10. The method of claim 9 wherein said storing further comprises:  
loading a program file when referenced during execution of said main program unit;  
storing said program file to said library file when said program file is an application  
program file; and  
determining whether execution of said main program unit has terminated.

11. The method of claim 10 wherein said library file further comprises a compressed file.

12. The method of claim 8 wherein said program files comprise Java™ class files and  
Java™ archive files.

13. The method of claim 12 wherein said system program file input comprises the Java™  
Development Kit.

14. The method of claim 13 wherein said library file comprises a Java™ archive file.

15. The method of claim 13 wherein said optimizing further comprises:

receiving an application program pathname of a referenced program unit in said  
5 library file;

determining a referencing program unit when said pathname is unexpected, said

referencing program unit referencing said referenced program unit; and

modifying said referencing program unit to remove any reference to said referenced  
unit.

16. A method for testing a dynamically loaded program, said program including a main  
program unit, said method comprising:

specifying a list including at least one application program file to be tested;

creating at least one library file containing only application program files loaded  
during execution of said main program unit; and

indicating incomplete test coverage when at least one file in said list is not represented  
in said library file.

17. The method of claim 16 wherein said creating further comprises:

20 specifying a first at least one pathname for system program files;

specifying a second at least one pathname for application program files;

executing said main program unit using said first at least one pathname and said  
second at least one pathname for dynamically loaded program files; and  
storing each application program file loaded during execution of said main program  
unit to said library file.

5

18. The method of claim 17 wherein said storing further comprises:

loading a program file when referenced during execution of said main program unit;  
storing said program file to said library file when said program file is an application  
program file; and  
determining whether execution of said main program unit has terminated.

10

19. The method of claim 18 wherein said library file further comprises a compressed file.

20. The method of claim 16 wherein said program files comprise Java™ class files and  
Java™ archive files.

15

21. The method of claim 20 wherein said system program file input comprises the Java™  
Development Kit.

22. The method of claim 21 wherein said library file comprises a Java™ archive file.

20

means for executing said main program unit a first time;

5 loaded during said first execution of said main program unit;

means for executing said main program unit a second time using said system program file input and said at least one library file for dynamically loaded program files.

means for creating at least one library file containing only application program files loaded during execution of said main program unit; and

means for optimizing said program based upon a list of application program files in said library file.

means for specifying a list including at least one application program file to be tested;

means for creating at least one library file containing only application program files loaded during execution of said main program unit; and

means for indicating incomplete test coverage when at least one file in said list is not represented in said library file.

26. A program storage device readable by a machine, embodying a program of  
 5 instructions executable by the machine to perform a method to execute a dynamically  
 loaded program, the method comprising:  
 executing said main program unit a first time;  
 creating at least one library file containing only application program files loaded  
 during said first execution of said main program unit;  
 specifying a system program file input; and  
 executing said main program unit a second time using said system program file input  
 and said at least one library file for dynamically loaded program files.

27. The program storage device of claim 26 wherein said creating further comprises:  
 specifying a first at least one pathname for system program files;  
 specifying a second at least one pathname for application program files;  
 executing said main program unit using said first at least one pathname and said  
 second at least one pathname for dynamically loaded program files; and  
 storing each application program file loaded during execution of said main program  
 20 unit to said library file.

28. The program storage device of claim 27 wherein said storing further comprises:  
loading a program file when referenced during execution of said main program unit;  
storing said program file to said library file when said program file is an application  
program file; and  
5 determining whether execution of said main program unit has terminated.

29. The program storage device of claim 26 wherein said library file further comprises a  
compressed file.

30. The program storage device of claim 26 wherein said program files comprise Java™  
class files and Java™ archive files.

31. The program storage device of claim 30 wherein said system program file input  
comprises the Java™ Development Kit.

32. The program storage device of claim 31 wherein said library file comprises a Java™  
archive file.



33. A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method to optimize a dynamically loaded program, the method comprising:

creating at least one library file containing only application program files loaded

during execution of said main program unit; and

optimizing said program based upon a list of application program files in said library file.

34. A program storage device readable by a machine, embodying a program of instructions executable by the machine to perform a method to test a dynamically loaded program, the method comprising:

specifying a list including at least one application program file to be tested;

creating at least one library file containing only application program files loaded

during execution of said main program unit; and

indicating incomplete test coverage when at least one file in said list is not represented in said library file.

35. A method for representing a library file, said method including:

storing in at least one program unit field the pathname of every program unit loaded

during the execution of a dynamically loaded program, said dynamically loaded program including a main program unit; and

storing in a main unit field the pathname of said main program unit.

36. The method of claim 35 wherein

said program unit field said main unit field are contained within a JAR file and

said main unit field comprises a manifest file.

5

001E2D" 08062960